| University of Central Lancashire logo  CO1301 Games Concepts | 2021 - 2022 |
| --- | --- |
| Assessment 1 Report | |

*This form should be submitted together with your Assessment 1’s .cpp file and demo video. You* ***MUST*** *fill this form using the provided input spaces. Keep your answers clear and succinct and ensure the form does not overflow to a second page.*

|  |  |  |  |
| --- | --- | --- | --- |
| ID Number | 20862007 / CE | Full Name | Adnan Ali Mumtaz |

For each feature from the assessment brief listed in the table below, check its associated radio button if you have implemented the feature, or leave the radio button unchecked if you have not implemented the feature.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0 – 49** | | **50 – 59** | | **60 -69** | | **70 - 100** | |
| Basic Scene |  | Updated Camera |  | Cube Model Array |  | Randomised (Re)spawn |  |
| Sphere Movement |  | Points |  | Sphere to Box |  | Hyper Mode |  |
| Camera Movement |  | Scale Player |  | GameWon State |  | NPCs |  |
| Pause |  |  |  |  |  | Points Ranking |  |
| Quit |  |  |  |  |  |  |  |
| Box Pickup |  |  |  |  |  |  |  |

1. For each of the two camera types specified in this assessment, give 2 examples of game genres that utilise them.

|  |
| --- |
| Rpg and Strategy |

1. How is the distance check between the sphere and cubes implemented? Briefly describe the calculation involved.

|  |
| --- |
| The distance is checked between sphere and cubes using Sphere-to-Box collision detection method. This method consists of Point-to-Box and Point-to-Sphere collision detection.  In this method XMin, XMax, YMin, YMax, ZMin, ZMax are calculated with the position of an object and being subtracted from the Mins and Maxis getting added into the collision objects’ radius.  E.g.  objectXMin = (cube->GetX() – XMin) – CollisionObjectRadius.  objectXMax = (cube->GetX() + XMax) + CollisionObjectRadius.  This formula has been used for each cube’s X, Y, Z axis. Once the values of “objectX,Y,ZMin and “objectX,Y,ZMax” have been calculated. The simple Point-to-Box collision detection is used.  Its formula is:  if ( X > objectXMin && X < objectXMax && Y > objectYMin && Y < objectYMax && Z > objectZMin && Z < objectZMax )  {  }  X = collision objects x axis position  Y = collision objects y axis position  Z = collision objects z axis position  Overall, the benefit of this collision detection is that it come across far more smother than the usual way of collision, in which vectors and their length is calculated. |

1. Provide a state transition diagram to describe the behaviour of your camera. (Paste an image in the box below)

|  |
| --- |
| Diagram  Description automatically generated |

1. Describe the steps and calculations involved in ensuring spawned cubes are more than 10 units apart.

|  |
| --- |
| First of all, the vector and vectors length are measured between all cubes and sphere. This vector and their lengths have been measured using for loop to condense the program. Therefore, nested for loop checks the vectors by subtracting one cubes’ position from one’s cubes position.  Once vectors are calculated, the square values are calculated and then sum of square is calculated. Ultimately, using the if statement the code changes the position of the cube if it’s within the 10 units of another cube. |

1. Describe the steps and any calculations used in implementing NPC movement.

|  |
| --- |
|  |